# SpaceGlue: Linking Spaces for Adaptive and Situational Service Location

Tomoko Itao[1], Satoshi Tanaka[1], and Tatsuya Suda[2,3]*

[1] NTT Network Innovation Laboratories
3-9-11 Midori-cho, Musashino-shi,Tokyo, 180-8585 Japan
{tomoko, satoshi}@ma.onlab.ntt.co.jp

[2] Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425 USA
suda@ics.uci.edu

**Abstract.** We propose and describe a networking technology called SpaceGlue for locating, communicating with, and interacting with services/people in a ubiquitous computing environment. In SpaceGlue, service components are embedded in a local communication area called a *ubiquitous space* and collaboratively provide an application. A user can locate desired service components offered in the local space by sending a query within the space. To allow users to discover service components that match their preferences in remote spaces, SpaceGlue dynamically links or "glues" together different spaces based on relationships among spaces that are estimated from the behavior history of many users. For example, if many users often visit a cafe and theater on the same day, these two spaces creates bonds to each other, reflecting the strong relationship among them. This lets users in the theater discover services in the cafe. We propose an algorithm for manipulating bonds to enable adaptive service location. We designed and implemented SpaceGlue using a distributed service platform called Ja-Net and showed that SpaceGlue is useful for adaptively locating services through simulation.

## 1 Introduction

The recent expansion of wireless network coverage opens the door to ubiquitous computing applications — diverse service components will be embedded in the user's physical environment and integrated seamlessly with the user's local activities. We call such a service environment a *ubiquitous space* (hereafter, space). In these spaces, service components are more or less tied to real-world entity such as people, places (e.g., school, cafe and theater), environments (e.g.,

* A part of Tatsuya Suda's research presented in this paper was supported by the National Science Foundation through grants ANI-0083074 and ANI-9903427, by DARPA through Grant MDA972-99-1-0007, by AFOSR through Grant MURI F49620-00-1-0330, and by grants from the University of California MICRO Program, and Nippon Telegraph and Telephone Corporation (NTT).

temperature, humidity and air), things (e.g., furniture, food and plants), and information shared by people. Thus, ubiquitous spaces and their applications are situational and tend to exhibit strong locality. End users (hereafter, users) travel among multiple spaces and receive services that are offered locally in a space. Although navigating users through spaces and services that meet their needs is important, due to the magnitude and dynamism of spaces, estimating and suggesting spaces and services that best support users is not straightforward. To achieve this, an infrastructure for spaces must be capable of (1) collecting and analyzing pieces of information that are derived from real-world users' activities, (2) discovering useful rules (e.g., characteristic user behavior patterns) based on which services can be customized, (3) delivering resulting services to users' computing environments.

SpaceGlue is a technology for adaptively linking or "gluing" two spaces based on user behavior patterns to adaptively locate, communicate with, and interact with groups of services/people in different spaces. SpaceGlue dynamically creates/adjusts logical links called "bonds" between two spaces by estimating important relationships among spaces by collecting and analyzing users' behavior history information. Spaces tied with strong bonds are closely related and encouraged to collaborate with each other to jointly provide services to users. This promotes efficient user navigation among spaces and services that meet their preferences.

In this paper, we describe the model, design, and implementation of SpaceGlue. We implemented SpaceGlue using a distributed service platform called Ja-Net, which is characterized by autonomous service components and a relationship mechanism to dynamically establish logical links between service components. The adaptation capability of SpaceGlue is evaluated through simulation.

## 2 Model and Algorithms of SpaceGlue

### 2.1 Model of Service Location

A space is a unit of physical area such as a cafe, theater, or bookstore where wireless communication is available. PCs, PDAs, and mobile phones are nodes in a space. Each space defines a multicast group where all nodes in the space belong to the multicast group.

Various service components are provided and running on nodes. Different spaces may provide different service components in the context of each space. For example, a cafe waiter service, which takes an order and serves a cup of coffee to a customer, may be provided at a cafe while a movie trailer service, which delivers a movie content to customers, in a theater.

In SpaceGlue, a space is represented by a service component called a "space agent (SA)" where nodes that run SAs can communicate via the Internet. SAs dynamically create bonds with one or more partner SAs reflecting behavior patterns of many users (See Figure 1). For example, if the majority of users visit and buy something at a cafe and gift shop in the same day, SAs in the cafe and gift shop create bonds with each other. A bond can be thought of as an information cache that SA maintains for each of its partner SA. Its attributes include name, pointer to the partner, and strength that indicates the usefulness of the partner. Each SA creates a bond for each of its partners.

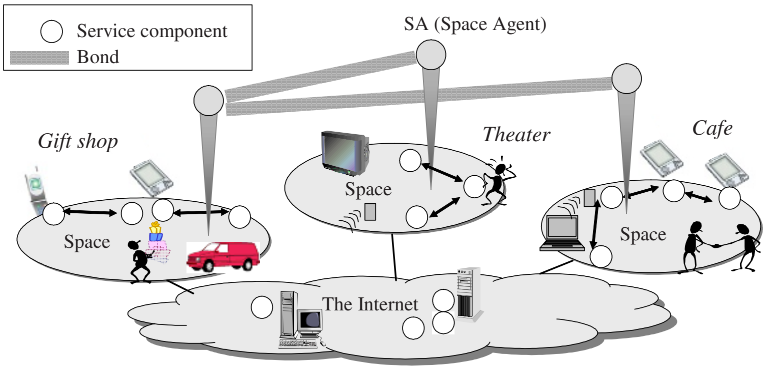


**Fig. 1.** Ubiquitous spaces are dynamically linked by bonds.

In SpaceGlue, a service component is known by its *advertisement* that includes a metadata description of the service it offers such as name and type and interface information to access the service. Each service component registers its advertisement with a local SA. SA maintains a list of advertisements (called a *local advertisement list*). To share local advertisements with other spaces in a meaningful manner, an SA sends its local advertisement list to remote SAs to which it has a bond. The receiver SA registers the received advertisement list (called a *remote advertisement list*) along with the sender SA's name.

To locate a service, a user first sends a query to its local SA, specifying advertisement attributes if necessary. Upon receipt of a query, the SA first examines its local advertisement list and then examines its remote advertisement lists. Remote advertisements are sorted in descending order of the bond strength and examined until a maximum of $M$ remote advertisements match the query. Finally, the SA replies to the user with a list of advertisements that matched the query. When a user receives an advertisement for a remote space and wishes to receive the service, the user moves to the space before receiving the service.

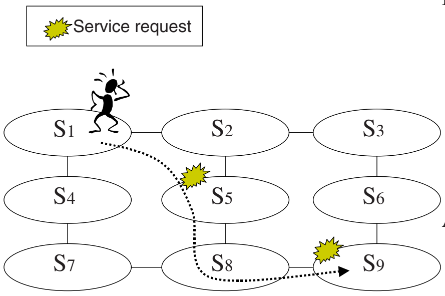
## 2.2 Bonding Algorithm

In SpaceGlue, as bonds are developed, advertisements propagate on a bond network. The bond strengths are dynamically adjusted, which enables spaces to always offer relevant advertisements reflecting behavior patterns of many users. This happens in the following manner.

**Adding history.** In SpaceGlue, each user maintains two types of history: *Information History (IH)* and *Action History (AH)*. A record of IH and AH is a key-value pair where the key is the name of a space that sent advertisements to a user and the value is a list of names of remote spaces that have a bond with the space specified as a key and whose advertisements were sent to the user via the space.

– IH is intended to record all advertisements that a user has ever received and that may have influenced the user's actions. When a user receives a query reply with a set of advertisements from a local space, a new record is added to the user's IH.
– AH is intended to record user actions that actually took place. When a user sends a request to an advertised service, a new record is added to AH. The name of the space that offers the requested service is set as a key along with a list of names of spaces whose advertisements were sent to the user via the space as a value.

For example, in Figure 2, suppose that a user visits spaces $S_1$, $S_5$, $S_8$, and $S_9$ and receives advertisements about remote spaces $S_2$ and $S_4$ (at $S_1$), $S_2$ and $S_8$ (at $S_5$), $S_5$, $S_7$ and $S_9$ (at $S_8$), and $S_6$ and $S_8$ (at $S_9$). At this point, records with keys with $S_1$, $S_5$, $S_8$, and $S_9$ are added to the user's IH. Suppose also that the user requested services that are offered by $S_5$ and $S_9$. Two records with keys $S_5$ and $S_9$ are added to the user's AH.



Information History (IH)

| Key | Value |
|---|---|
| S1 | S2 S4 |
| S5 | S2 S8 |
| S8 | S5 S7 S9 |
| S9 | S6 S8 |

Action History (AH)

| Key | Value |
|---|---|
| S5 | S2 S8 |
| S9 | S6 |

**Fig. 2.** Examples of information and action histories of user X, who visited spaces $S_1$, $S_5$, $S_8$, and $S_9$ and requested services at $S_5$ and $S_9$.

Let us denote the maximum length of IH as $N_{IH}$ and that of AH as $N_{AH}$. In our algorithm, the minimum values of $N_{IH}$ and $N_{AH}$ are 2 and 1. When the history exceeds the maximum length, the oldest record is deleted first.

**User evaluation.** When a user makes a query and receives advertisements in a space, the user evaluates them based on the level of satisfaction for each advertisement in one of three ways: (1) *positive*: a user requests an advertised service (moves to the space if the advertisement is forwarded from a remote space), (2) *neutral*: simply ignores or closes advertisements, and (3) *negative*: explicitly rejects the advertisement. Upon receiving such indication of satisfaction level, SAs create/adjust bonds among SAs by looking at the history of individual user behaviors (See Section 2.2).

**Creating glue.** Assume that a user is currently in space $S_{current}$. Let us denote the $i$ th key (space) in the user's AH as space $S_i$. When a user requests a service in $S_{current}$ (i.e., positive reaction), SpaceGlue examines the user's AH and creates a bond between $S_{current}$ and $S_i$ as follows:

$$\begin{array}{l} \quad create\ (S_{current},\ S_i) \\ if\ \ bond\ (S_{current},\ S_i) = \emptyset \\ where\ \ AH\ \ni\ record(key = S_i) \end{array}$$

For example, in Figure 2, when a user who had once requested a service in $S_5$ requests a service in $S_9$, a bond is created between $S_9$ and $S_5$.

**Adjusting bond strengths.** Assume that a user is currently in space $S_{current}$. Let us denote the $i$ th key (space) in the user's IH as space $S_i$. When a user in $S_{current}$ indicates positive, neutral, or negative reaction to an advertisement that he/she received in $S_{current}$, SpaceGlue examines the user's IH and updates the bond between $S_{current}$ and $S_i$ as follows:

$$\begin{array}{l} \quad update\ (S_{current}, S_i) \\ if\ \ bond\ (S_{current},\ S_i)! = \emptyset \\ where\ \ IH\ \ni\ record(key = S_i,\ value \ni S_{current}) \end{array}$$

Bonds between spaces $S_{current}$ and $S_i$ is strengthened (weakened) if the user indicates positive (negative) reaction to the service offered in $S_{current}$ and the user had received an advertisement about a service offered in space $S_{current}$ in space $S_i$. For example, in Figure 2, when a user requests a service in $S_9$ and indicates positive satisfaction, the bond between $S_9$ and $S_8$ is strengthened.

## 2.3 AdAppli: An Example Service Scenario

In this section, we illustrate a realistic application scenario using SpaceGlue. The application is called *AdAppli* (Advertisement Application). In the scenario, we assume a shopping mall complex with several shops. Each shop corresponds to a space and runs an SA on a shop PC. A number of customers visit the mall, spend time shopping, and then leave. We assume that many of the people who visit the mall carry a PDA, and they run user agents representing themselves. Each shop stores a single advertisement about the shop's merchandise (in a local advertisement list). It also stores advertisements of other shops that are linked with glue (in a remote advertisement list). Customers who visit a shop can retrieve these advertisements by sending a query. In AdAppli, the content of advertisements retrieved in a shop adapts to typical shopping sequences of a crowd of customers (users) as described below.

**Trends generated in user preferences.** In AdAppli, we consider the following assumptions about customer preferences: Different customers may have different preferences for types of shops and merchandise, and each of them buys merchandise that matches his/her preferences. Although different customers have different preferences, we assumed that the differences were small, and partly overlapped. This results in a non-uniform distribution of customer preference for merchandise, i.e., trends in the preferences of the crowd in the mall. We further assumed that although customers constantly come and go in the mall, the types of shops and merchandise that they prefer do not change significantly over a certain time period.

**History management.** In AdAppli, shops in the shopping mall create and acquire IH and AH as follows.

– A shop sends advertisements to the PDAs of customers in the shop. By looking at advertisements on user's PDA, shop SA knows which advertisements the customer have ever received (i.e., the IH of the customer).
– A shop issues a coupon that is available at all shops in the shopping mall to a customer who bought something in the shop. By looking at coupons on user's PDA, shop SA knows which shop the customer made purchses at (i.e., the AH of the customer).

**User satisfaction level.** In AdAppli, the level of user's satisfaction is mapped to the user's actions as follows:

– User buys merchandise by looking at an advertisement, which indicates that the user is *positive* about the advertisement.
– User simply ignores or closes an advertisement, which indicates that the user is *neutral* about the advertisement.
– User explicitly indicates that he/she dislikes the advertisement, which indicates that the user is *negative* about the advertisement.

**User navigation process.** Assume that bonds have not yet been established among shop SAs and that user navigation based on SpaceGlue happens in the following manner.

When customer A first arrives at the mall, she does not go in any specific direction but enters shops at random. When customer A arrives at a cafe, for example, the cafe's CE displays its own advertisement which contains information about its merchandise (e.g., the shop's coffee menu) on customer A's PDA.

Customer A may buy merchandise (e.g., a cup of coffee) if it matches her preferences. When this happens, the cafe's SA issues her with a coupon which she may use when she next buys something at another shop.

When she leaves the shop and moves to the next shop at random (e.g., gift shop) in the mall, the shop's SA displays its own advertisement on her PDA. This time, customer A may buy a gift using the coupon issued at the cafe. When the gift shop's SA receives the coupon, it attempts to create a bond with the cafe's SA. This will cause the gift shop's SA and cafe's SA to display each others advertisements on customer PDAs. Thus, when a new customer arrives at the cafe, ads of the cafe and the gift shop will be displayed on her PDA, and vice versa.

In AdAppli, a number of different customers visit different shops, based on which shop SAs have established bonds and the bond strengths are adjusted according to customers' shopping pattern. Through these bonds, shop SAs can guide customers through the mall by having their partners show their advertisements to potential customers. Without these bonds, customers would have to move randomly between shops.

## 3 Design and Implementation of SpaceGlue on Ja-Net

We implemented SpaceGlue for the AdAppli scenario using the Jack-in-the-Net Architecture (Ja-Net) [1][2][3] that enables adaptive service creation/provision. In Ja-Net, application services are implemented by a group of autonomous service components (i.e., agents) called *cyber-entities* (CEs). Each CE implements one or more service components and follows simple behavioral rules (such as migration, replication, energy exchange, death, and relationship establishment with other CEs). They autonomously form organizations to collectively provide higher services by establishing relationships with several other CEs. Ja-Net platform software provides a runtime environment for CEs called ACERE (ACE Runtime Environment) and APIs to manipulate relationships among CEs, such as creating, deleting, and updating relationships, and selecting CEs based on attributes of CEs or their relationships. Thus, the bonding algorithm of SpaceGlue can be easily implemented based on the relationship mechanism of Ja-Net.

### 3.1 System Overview

Figure 3 shows a system overview of SpaceGlue. Shops 1 and 2 are defined as spaces. PCs that run SAs of shops 1 and 2 are connected to the same network via Ethernet. We used a notebook PC instead of PDA as a user's terminal on which a user CE is running. Each shop is coupled with an IEEE802.11b access point allowing wireless communication with users' notebook PCs. Although the coverage of multicast messages in Shops 1 and 2 overlaps, multicast messages are distinguished by the MAC address of access points that are piggybacked on each message. A user carries a notebook PC and can move between Shops 1 and

2. A daemon process is running on the user's notebook PC to monitor the link status and notify the Ja-Net platform software whether the link is up or down.

The shop and user GUI windows which are shown on the shop's PC and user's notebook PC are also shown in Figure 3. On the user's notebook PC, an advertisement is shown along with buttons "buy", "close", and a checkbox "don't like". If the user selects "buy" button, it is interpreted as a positive reaction. If the user marks the checkbox and selects any button, it is interpreted as a negative reaction. Otherwise, it is interpreted as a neutral reaction.

The bond strength is adjusted by

$$S = \frac{P}{T}, \tag{1}$$

where $P$ is the number of positive reactions and $T$ is the number of positive or negative reactions that the Shop CE received from users when it sends the advertisement of a partner Shop CE. The initial value of strength is set to 1.

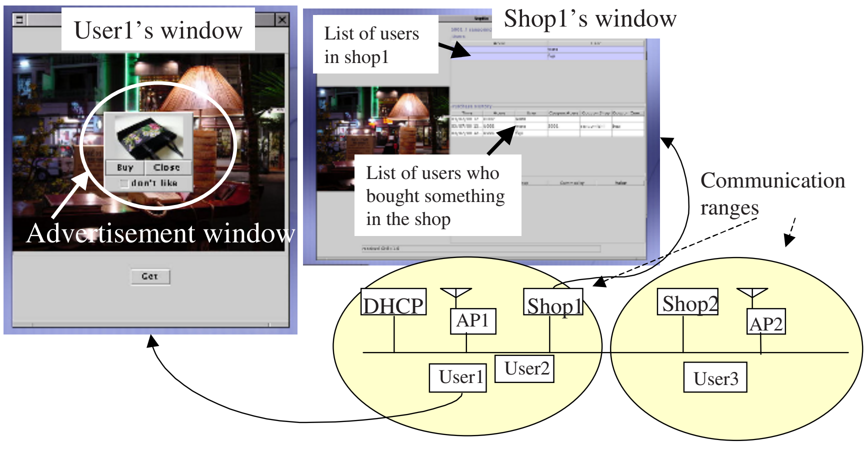


**Fig. 3.** Configuration of GUI windows on shop's PC and user's notebook PCs.

## 3.2 Collaboration

Shop SA and users are implemented as CEs (i.e., *Shop CE* and *User CE*). In addition, we implemented the following two functions/information content as CEs to enable active collaboration.

– *Advertisement CE* carries an advertisement of a Shop CE and definition of GUI components (such as buttons and panels) to display on the user's notebook PC.

– *UserTerminal CE* controls the GUI of the user's notebook PC such as displaying advertisements and inputting user commands.

In our design, IH and AH are represented by Advertisement CEs and coupons that a User CE has. By looking at the Advertisement CEs on a user's notebook PC, a Shop CE knows the IH of the user. Similarly, by looking at coupons carried by a User CE, Shop CE knows the AH of the user. Note that coupon information includes the name of the coupon's originator SA. Note also that our implementation is designed for IH and AH with history lengths of 2 and 1, respectively.

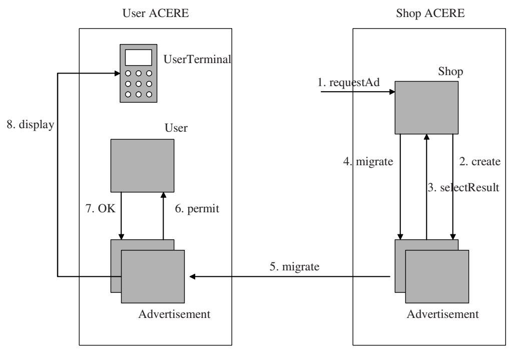


**Fig. 4.** Collaboration for sending advertisements

Collaboration among CEs in sending advertisements is illustrated in Figure 4. The sequence is described below:

1. Upon arrival in a space or receipt of a query issued by a human user, the user CE sends a query to local shop CE.
2. The shop CE creates an Advertisement CE.
3. The ID of the Advertisement CE is reported to the shop CE via selectResult event.
4. The shop CE sends a migration request to the Advertisement CE.
5. The Advertisement CE migrates to the user CE's ACERE. (adding IH)
6. The Advertisement CE asks permission to display its content on the display.
7. The user CE gives permission.
8. The Advertisement CE sends request to a UserTerminal CE to display its content on the display.

Collaboration among CEs for relationship (bond) creation is illustrated in Figure 5. The sequence is described below:

1. A human user buys merchandise (clicks the "buy" button of an advertisement), which is reported to the User CE.

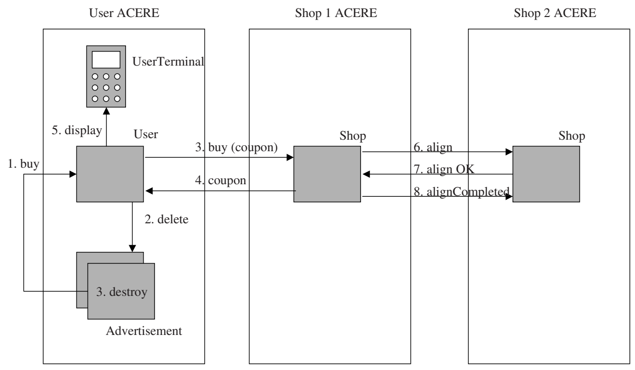


**Fig. 5.** Collaboration for creating glue (relationships)

2. The User CE deletes the Advertisement CE.
3. The User CE informs Shop1 CE of the purchase. A coupon issued by Shop2 CE at a previous purchase is sent to Shop1 CE along with the message.
4. The Shop 1 CE issues a coupon and sends the information to the User CE. (adding AH)
5. The User CE displays on the user's notebook PC that it has received a coupon.
6. By looking at the coupon information sent by the User CE, Shop1 CE asks Shop2 CE to create a bond.
7. Shop2 CE agrees and creates a bond with Shop1 CE.
8. Shop1 CE creates a bond with Shop2 CE.

## 3.3 System Measurement

The program sizes of Shop CE, User CE, Advertisement CE, and UserTerminal CE are 222, 52, 282, and 194 KB, respectively. The Ja-Net platform software is 388 KB. The number and size of messages sent among CEs during the sequence described in Section 3.2 (i.e., sending advertisements from Shop CE to user and creating glue between Shop CEs) is shown in Table 1. Ja-Net messages are 4.5 times the number of SpaceGlue messages.

**Table 1.** Messages sent while a user is in a shop

| Message | Num. messages | Total message size | Average message size |
|---|---|---|---|
| Ja-Net | 9 | 37 KB | 14 KB (for migration), 1.3 KB (excluding migration) |
| SpaceGlue | 2 | 6 KB | 3.3 KB |

## 4 Simulations

In SpaceGlue, we expected that as the bonds among shop SAs develop, some shop SAs would become capable of efficiently sending them to potential customers by showing their advertisements to customers in their partners' shops. This would lead to a situation where customers tend to move to other shops with which the current shop has a bond (influenced by the advertisements received in the current shop), and groups of customers with the same preference move among the same set of shops. We examined these features and evaluated the adaptation capability in SpaceGlue. We conducted several simulations by (1) running multiple user CEs and shop CEs on a single PC and (2) using a simulator of SpaceGlue. Simulation results described in Section 4.2 were obtained through (1) and those in Sections 4.3, 4.4, and 4.5 through (2). Note that in (1), User CEs automatically simulate human shopping activity, i.e., visiting shops, buying something, and evaluating advertisements. Note also that (2) yielded equivalent results to (1).

### 4.1 Simulation Definitions

**Customer preferences.** In the simulation, each shop had an ad keyword representing the shop and its merchandise. Each user had a set of five distinct ad keywords out of $M$ ad keywords in its preference keywords. Customers bought merchandise if the ad keyword of target shop matched one of their preference keywords at the probability of $P_P$ (this is called the *Purchase Probability*, i.e., the likelihood that the user will buy merchandise).

**Significance in trends in customer preferences.** We used mutual information [4] to measure the significance of trends generated by customer preferences in the shopping mall. The calculation of mutual information was based on the co-occurrence of an ad keywords pair in the preference keyword of all customers, and is defined as follows:

$$I(A;B) = \sum_{A,B} P(a,b) log \frac{P(a,b)}{P(a)P(b)}, \tag{2}$$

where $P(a,b)$ is the probability of co-occurrence of two *ad keywords* in the preference keywords for all customers, $a$ and $b$, and $P(a)$ and $P(b)$ are the probability of the single occurrence of an ad keyword $a$ and $b$ in the preference keywords for all customers, respectively. The greater the $MI$ value is, the more significant the trends in customer preferences are.

To determine how the preferences of a group of customers affect the performance of AdAppli, we prepared two preference keyword sets with different $MI$ levels: $MAX$, $High(H)$, $Low(L)$, and $MIN$ where $MAX$ and $MIN$ were maximum and minimum mutual information, respectively. Figure 6 shows an example of ad keywords occurring (depicted as 1–36 on the $x$ axis) in preference keywords when $MI = MAX$ (i.e., the most significant trends in the preferences of the crowd).

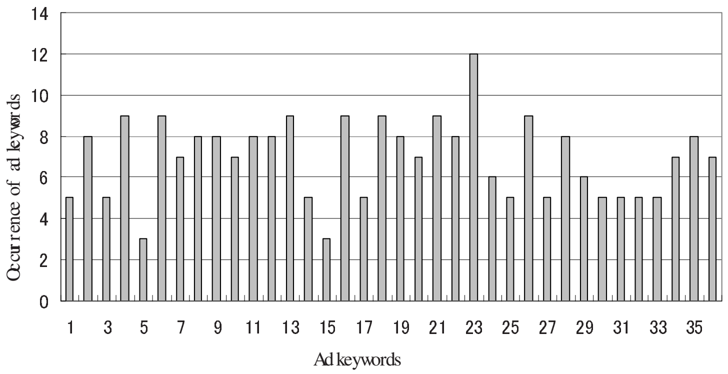


**Fig. 6.** Example of preference keywords occurring with $MI = MAX$.

**Evaluation measure.** The *Recall* and *Precision* of each Shop CE are defined as:

$$Recall = \frac{C}{T} \cdot 100, \quad \text{and} \tag{3}$$

$$Precision = \frac{C}{H} \cdot 100, \tag{4}$$

where $C$ is the number of potential customers for a given shop (i.e., User CEs that have the ad keyword of the Shop CE in their preference keywords) that the Shop CE sent its ad to, $T$ is the total number of the given shop's potential customers, and $H$ is the total number of customers that the Shop CE sent its ad to. $C$ and $H$ were obtained by examining the value of user reactions received by a given shop during five periods when ads were broadcast. $C$ is the number of users' positive or neutral reactions and $H$ is the total number of users' reactions. Recall and precision tend to have a trade-off relationship, but we need to achieve both high recall and precision.

To evaluate the adaptation capability of SpaceGlue, we used the *F-measure* [4] which combines recall and precision in a single efficiency measure (the harmonic mean of precision and recall), which is given by

$$F{-}measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \cdot 100 \tag{5}$$

## 4.2 Adaptation Capability of SpaceGlue

In our simulation, we ran 36 Shop CEs and 50 User CEs. Each User CE had five distinct ad keywords out of 36 ad keywords in its preference keywords. *Purchase probability* $P_p$ was set to 0.7 for all user CEs. Initially, User CEs were randomly placed in one of the 36 shops. Each shop CE periodically broadcast a set of ads $N$ times. For each shop CE, the time for broadcasting ads was set to 90 seconds.

Figure 7 shows the time variance of F-measure values at each $MI$ level. Here, we measured the recall, precision, and F-measure for each shop CE and took the average. The $x$ axis in those figures are the respective times that ads were broadcast ($N$). As shown in the graph, the maximum F-measure values achieved depend on the $MI$ value, i.e., significance of trends in user preferences. The more significant the trends, the greater the F-measure value. For example, after developing enough bonds, the F-measure exceeded 80 when trends were most significant ($MI = MAX$). That is, 80% of customers were satisfied with advertisements that they received at each shop. Thus, we may conclude that SpaceGlue is effective at customizing the content of advertisements in AdAppli.

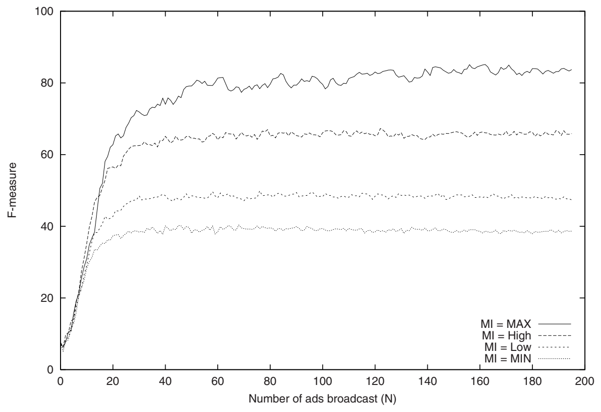


**Fig. 7.** F-measures with $MI = MAX, H, L, MIN$.

Figure 8 shows the time variance of the number of bonds that shop CEs created at $MI = MAX$ where the $x$ axis in these figures is the respective times that ads were broadcast ($N$). The number of bonds increased in the beginning as time progressed and saturated after $N = 140$. Note that when the F-measure value exceeded 80 for the first time at $N = 52$, the number of bonds was 4.2, which was less than the maximum number of bonds created during the simulation (5.5). That is, a subset of total bonds may be sufficient to achieve the optimal F-measure.

## 4.3 Impact of Glue

In order to examine the impact of glue, we compared the F-measure values in the following three cases: (1) SpaceGlue, (2) Flooding, (3) No glue. In (1) “SpaceGlue”, shop CEs sent advertisements using SpaceGlue technology. In (2) “Flooding”, each and every shop CE sent advertisements of all other shops to customers in their shops. Thus, customers received 36 advertisements because there were 36 shops in the mall. In (3) “No glue”, a shop CE sent a single

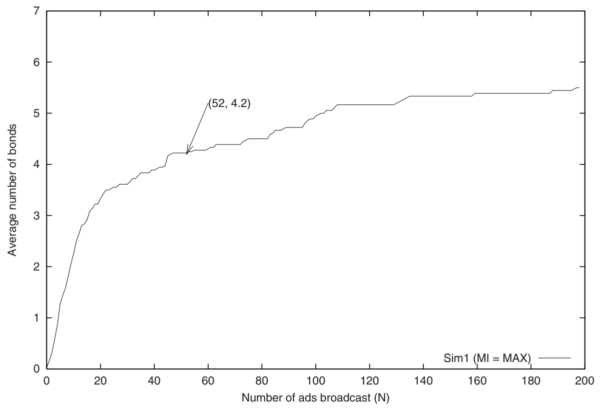


**Fig. 8.** Number of bonds with $MI = MAX$.

advertisement of its own to customers in the shop. Thus, customers move among shops at random.

The results are shown in Figure 9 where the time variance of F-measure in cases (1)–(3) are shown. Case "Flooding" and "No glue" achieved almost constant F-measure (about 23 and 5, respectively) throughout the simulation. Although "Flooding" achieved a better F-measure during the first 14 time periods, "SpaceGlue" overtook it once it developed bonds. The F-measure of "Flooding" is inferior to "SpaceGlue" because "Flooding" always achieves poor precision. In contrast, by adaptively selecting a subset of advertisements that match customer preferences, "SpaceGlue" achieves good precision. Thus, SpaceGlue achieves adaptability by its bonding technique.

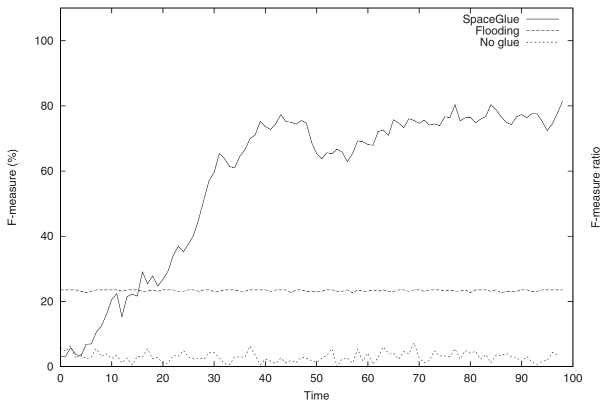


**Fig. 9.** F-measure comparison among SpaceGlue and flooding modes with $MI = MAX$.

### 4.4 Impact of History Length

To examine the impact of history lengths of IH and AH, we measured the average F-measure during 100 time periods with different history lengths in comparison. We compared the history length combinations of (IH, AH) where IH = 2, 4, 8 and AH = 1, 2, 4, 8. Figure 10 shows the results. The (IH, AH) = (4, 4) achieved the maximum F-measure on average (67) and was thus most appropriate. Note that (IH, AH) = (8, 8), which exhibited the maximum history length, achieved only 57. Thus, we may say that there is an optimal combination of history lengths of IH and AH. Examining such optimal history lengths through simulation is important in SpaceGlue.

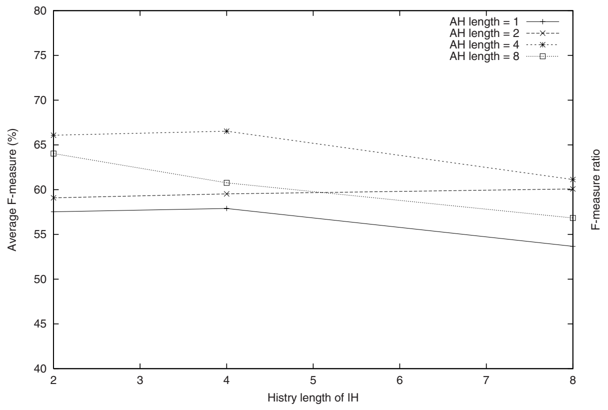


**Fig. 10.** Impact of history lengths on F-measures (average) with $MI = MAX$.

### 4.5 Scalability of SpaceGlue

Figure 11 shows average F-measure during 1000 time periods with different numbers of shops and customers; (shop, customers) = (36, 50), (100, 150), and (400, 600). The $x$ axis shows the number of shops (36, 100, 400). The line labeled "SpaceGlue" shows the average F-measure achieved using SpaceGlue. The line labeled "No glue" shows the average F-measure achieved without glue (no bond). The line labeled "SpaceGlue/No glue ratio" shows the ratio of average F-measure of "SpaceGlue" and "No glue". Although the F-measure values of "SpaceGlue" and "No glue" decreased as the number of shops increased, the ratio of F-measure between "SpaceGlue" and "No glue" increased. That is, SpaceGlue is effective at maintaining adaptability in a scalable manner.

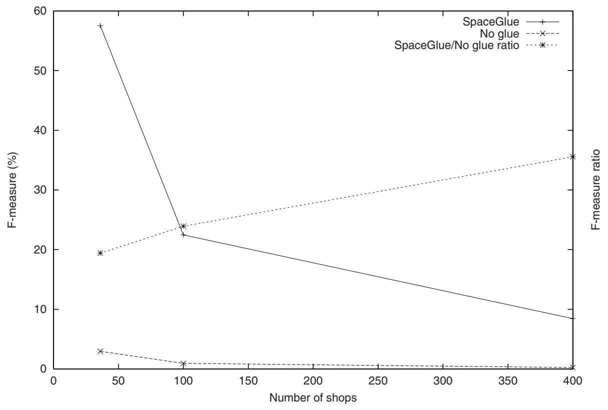


**Fig. 11.** Impact of the number of shops and customers with $MI = MAX$.

## 5 Related Work

The concept of SpaceGlue is similar to the association rule of data mining[6] although their mechanisms are different. Data mining is based on a centralized architecture where all the information is accumulated in a central database because the data mining algorithm requires entire sets of information as input. Unlike data mining, SpaceGlue is designed based on a distributed architecture where the bonding algorithm is implemented by each SA and processed based on local interactions among SAs. Thus, SpaceGlue requires no central administration. In addition, SpaceGlue provides a total environment for adaptive service provision: implementing, delivering, analyzing, and customizing services are all-in-one while data mining systems are operated separately from service systems. Thus, SpaceGlue is more portable and easy to deploy.

Although the current SpaceGlue only considers bonds between two CEs, further improvement of estimation should be possible if it considers topological information of bond networks. PageRank[7] provides a method of ranking network nodes based on the topology of hyperlinks between web pages. However, calculating PageRank requires topology data for each and every pair of web pages as input, which is difficult in the dynamic environment assumed in SpaceGlue where bonds are dynamically created/adjusted based on user activities. Thus, application of PageRank to SpaceGlue requires a distributed mechanism to calculate PageRank.

## 6 Conclusion

SpaceGlue is a technology for adaptively linking or "gluing" ubiquitous spaces to adaptively locate, communicate with, and interact with groups of services/people in different ubiquitous spaces. We are investigating SpaceGlye for mobile phone networks. For this, a lightweight version of Ja-Net platform software is necessary.

Various applications can be designed using SpaceGlue, including (1) Situational marketing: shop owners in a shopping mall identify potential customers based on their behavior patterns and send commercial advertisements that match their preferences at the right place and time, (2) Community creation: IM chatters can search for groups of users with related interests based on the association degree between spaces, and (3) Collaborative services: services in two related spaces may jointly provide a higher-level service to create new services. We will design and implement such applications to empirically verify that SpaceGlue is useful for building ubiquitous computing applications.

## References

1. T. Itao, T. Nakamura and M. Matsuo, T. Suda, and T. Aoyama, "Service Emergence based on Relationship among Self-Organizing Entities," Proc. of the IEEE SAINT2002, Nara, Japan, Jan. 2002. (Best Paper)
2. T. Itao, T. Nakamura, M. Matsuo, T. Suda, and T. Aoyama, "Adaptive Creation of Network Applications in the Jack-in-the-Net Architecture," Proc. of the IFIP Networking 2002, Pisa, Italy, May 2002.
3. T. Suda, T. Itao, and M. Matsuo, "The Bio-Networking Architecture: A Biologically Inspired Approach to the Design of Scalable, Adaptive, and Survivable/Adaptable Network Applications," to appear, Internet as a Large Complex System, the Santa Fe Institute Book Series, Oxford University Press.
4. C. J. Van Rijsbergen, "Information Retrieval", 2nd edition, Butterworths, London, 1979.
5. Web services web site, http://www.webervices.org/
6. R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. of SIGMOD, pp. 207–216, 1993.
7. PageRank web site, L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford Digital Library Technologies, Working Paper 1999–0120, 1998.